GPT-1

Improving Language Understanding by Generative Pre-Training

0. Abstract

- NLU(자연어 이해)는 textual entailment, question answering, semantic similarity, document classification 등의 과제를 포함
- unlabeled된 텍스트 코퍼스는 많지만, 특정 과제를 학습하는데 쓰이는 labeled 데이터는 희소함
- 위의 unlabeled 텍스트 코퍼스를 이용하여 언어모델을 generative pre-training하는 데 사용하고, 특정한 테스크에 맞게 discriminative fine-tuining하여 높은 성과를 냄
- 기존 기법과 달리 fine-tuning 과정에서 task-aware input transformations을 사용하여 효과적인 전이를 달성, 모델 구조의 최소한의 변화만 필요로 함
- 이러한 task-agnostic model이 discriminative training model 능가함

1. Introduction

- raw text로부터 학습하는 능력은 supervised learning에 대한 의존도를 줄이는 데 필수적
- 대부분의 딥러닝 기법은 대량의 라벨링된 데이터를 필요로 하며, 이러한 주석 데이터가 부족한 도메인에서는 적용이 제한됨.
- 비라벨링 데이터에서 언어적 정보를 활용할 수 있는 모델은 추가 데이터 수집 작업을 대체할 수 있는 효과적인 대안
- 충분한 supervision이 존재하는 경우, unsupervised learning을 통해 우수한 표현 학습을 수행하여 모델 성능 향상
  ex) pre-trained된 단어 임베딩의 광범위한 활용
- 비라벨링된 텍스트에서 풍부한 정보를 활용하는 것이 어려운 두가지 이유

1. 전이 학습에 유용한 텍스트 표현을 학습하기 위해, 어떤 optimization objective가 가장 효과적인지 명확하지 않음
2. 학습된 표현을 target task로 효과적으로 전이하는 최적의 방법에 대한 합의가 존재하지 않음

- unsupervised pre-training과 supervised fine-tuning을 결합한 semi-supervised approach를 통해 '언어 이해 과제'를 해결
- 최소한의 조정만으로 다양한 과제에 전이 가능한 보편적 표현을 학습하는 것이 목표
  ->대규모의 비라벨링된 텍스트 코퍼스와 라벨링된 훈련샘플(target tasks) 데이터셋을 이용하는 환경을 가정
  -> 위의 두가지 데이터셋이 동일한 도메인일 필요가 없다고 가정
- 두 단계로 학습 수행
  1st. 비라벨링 데이터에서 language modeling 목표를 사용하여 모델의 초기 가중치를 학습
  2nd. supervised objective를 통해 해당 모델을 목표 과제에 맞게 조정
- 모델 구조로는 structured memory를 제공하여 다양한 과제에서 강건한 전이 성능을 달성하는 Transformer 사용

- 전이 과정에서는 구조화된 텍스트 입력을 단일 연속 토큰 시퀀스로 처리하는 traversal-style approach를 기반으로 한 task-specific input adaptations 활용
  -> 이러한 입력 변환이 pre-trained된 모델의 구조를 최소한으로 변경하면서도 효과적인 fine-tuning을 가능하게 함

2. Related Work

Semi-supervised learning for NLP

- 위 연구는 sequence labeling이나 text classification과 같은 다양한 과제에 적용 가능함
- 초기에는 unlabeled data에서 단어, 구 수준의 통계를 계산하여 supervised model의 feature로 사용함
- unlabeled 코퍼스에서 학습된 word embeddings이 다양한 과제에서 성능 향상에 기여함
  -> 이러한 접근법은 주로 단어 수준의 정보만 전이
  <-> 본 연구는 상위 수준의 의미를 포착하는 것이 목표
- 최근 연구는 unlabeled 코퍼스를 이용해 학습된 phrase, sentence 수준의 임베딩은 다양한 target tasks에서 벡터 표현으로 인코딩하는 데 사용됨.

Unsupervised pre-training

- semi-supervised learning의 한 형태
- supervised learning 목표를 변경하는 대신 적절한 초기화 포인트를 찾는 것을 목적으로 함
- 이후 연구에서는 pre-trainig이 정규화 기법으로 작용하여 신경망의 일반화 성능 향상시킴
- 본 연구와 가장 유사한 연구들은 language modeling 목표를 사용하여 신경망을 사전학습한 후, 지도학습 방식으로 task에 fine-tuning 하는 방법
  -> but, LSTM 모델을 사용하여 문맥 정보를 학습하는 데 한계
  <-> 본 연구에서는 Transformer 네트워크를 활용하여 장기적인 언어적 구조 포착
- 다른 접근법으로는 사전 학습된 모델에서 생성된 hidden representations을 지도 학습 모델의 auxiliary feature로 사용하는 방식 취함
  -> but, 많은 수의 새로운 매개변수 필요로 함

Auxiliary training objectives

- auxiliary training objectives를 추가하는 것은 semi-supervised learning의 다른 형태
- 초기 연구에서는 다양한 보조 과제를 사용하여 semantic role labeling 성능 개선
- 이후 연구에서 auxiliary language modeling objective를 추가하여 sequence labeling 과제에서 성능 향상

3. Framework

GPT-1은 두 단계의 학습 절차를 가짐

1. 대규모의 텍스트 코퍼스를 통해 언어 모델을 학습
2. discriminative task에 맞게 labeled data를 사용하여 fine-tuning

3.1 Unsupervised pre-training

- unsupervised 말뭉치 토큰 $U = u_1, \ldots, u_n$ 이 주어질 때, 표준 language modeling objective는 다음의 likelihood를 최대화 하는 방향

$$L_1(\mathcal{U}) = \sum_i \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta) \quad (1)$$

- k 는 context window 사이즈, Θ는 뉴럴네트워크의 파라미터
- 이 네트워크에서 $u_{i-k}, \ldots, u_{i-1}$가 주어졌을 때 u_i의 확률값을 계산하는 식
  예컨대, **I love you**라는 문장이 있으면 **I, love** 가 주어졌을 때 **you**를 예측하는 확률값
- GPT-1은 다층 Transformer의 decoder 사용
- 입력 context tokens에 대해 multi-headed self-attention 연산을 적용한 후, position wise 피드포워드 레이어를 거쳐 출력 distribution 생성

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\ P(u) &= \texttt{softmax}(h_n W_e^T) \end{aligned} \quad (2)$$

- $U = u_1, \ldots, u_n$는 context vectorof tokens, n은 레이어의 개수, W_e는 토큰 임베딩 행렬, W_p 는 위치 임베딩 행렬

3.2 Supervised fine-tuning

- Unsupervised pre-trainingd으로 훈련된 모델의 파라미터를 가져와서 새로운 task에 맞는 labeled 된 데이터 C에 훈련시킴
- supervised learning이기 때문에 시퀀스에 따른 label 값 필요, input token $x^1, \ldots, x^m$을 $label = y$로 가정
- pretrain된 모델의 **Position-wise layer와 softmax layer 사이에 linear layer**(Wy) 하나를 추가하여 각 task마다 label y를 예측

$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y).$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

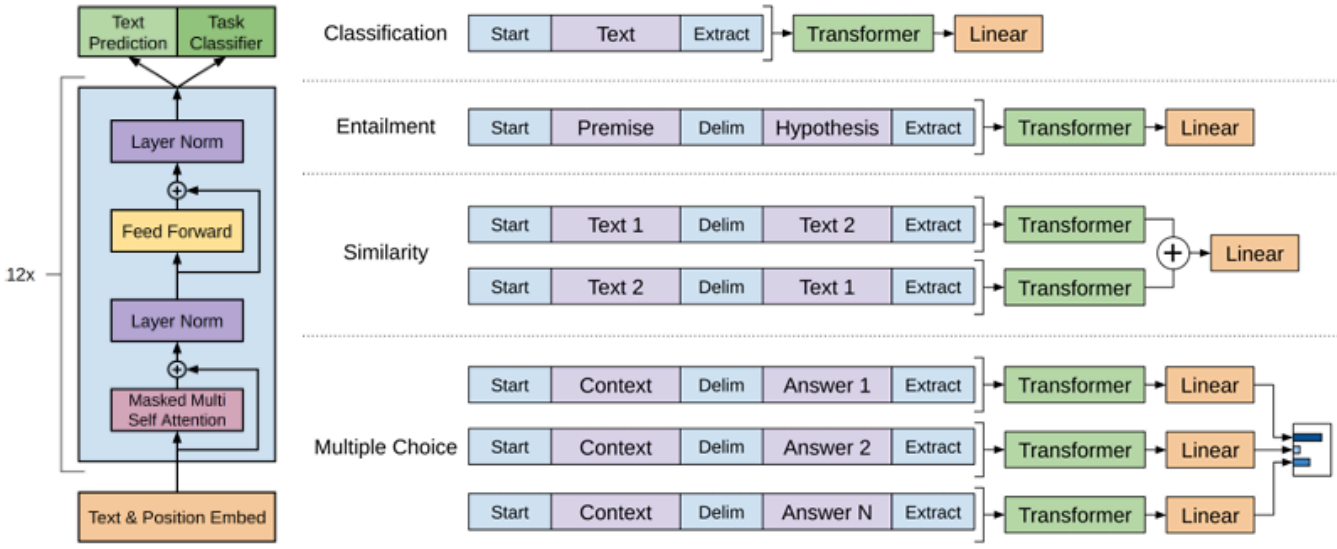
- Related work에서 fine-tuning 과정에서 language modeling을 auxiliary objective로 포함하여 성능을 높였는데 (a) supervisied model의 일반화 성능을 향상시키고, (b) 수렴 속도를 가속시킴
- 최종적으로 최적화할 손실 함수는 다음과 같이 정의

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

3.3 Task-specific input transformations

- question answering이나 textual entailment와 같이 구조화된 input이 필요한 작업에서는 입력 데이터가 정해진 순서대로 정리되어야 함
  -> GPT-1의 traversal-style 접근법을 사용하여 구조화된 입력을 순차적인 시퀀스로 변환하고, 이를 사전 훈

련된 모델이 처리할 수 있도록 함



- **Classification**: 입력되는 text 앞 뒤로 <s> <e> token을 부착시켜 Transformer의 입력으로 넣음
- **Textual entailment**: 전제(premise)와 가정(hypothesis) 두 가지의 시퀀스 토큰들을 중간 **구분 문자($)** 를 사용하여 한번에 네트워크에 forward 함
- **Similarity:** 두문장의 유사성을 비교할 때 어느 전제와 가정처럼 순서가 존재하지 않습니다. 그래서 문장 순서를 반영하기 위해 **(text1, text2), (text2, text1) 두 가지를 각각 모델에 forward 하여** 마지막 linear layer에 들어가기 전 **element-wise**로 합하여 출력함
- **Question Answering and Commonsense Reasoning:** 이 task에서는 지문 z, 질문 q, 정답 리스트 (a1,a2,...ak)가 있습니다. 각 정답 리스트 k개만큼의 각각 모델에 forward 하여 리스트들의 softmax를 구해 가장 정답에 가까운 값을 구함

5. Analysis

Impact of number of layers transferred

- 임베딩(embedding)만 전이해도 성능이 향상됨
- 각 Transformer 레이어를 추가할수록 성능이 점진적으로 향상됨
- MultiNLI에서는 모든 레이어를 전이(full transfer)하면 성능이 최대 9% 향상됨

Zero-shot Behaviors

- 생성 사전 훈련 중 제로샷 성능이 꾸준히 증가
  -> 생성 사전 훈련이 다양한 태스크에 유용한 기능 학습을 지원함을 의미
- LSTM은 제로샷 성능의 분산이 크며(variance가 높음), Transformer 구조가 전이에 더 유리함

Ablation studies

- fint-tuning 단계에서 보조 LM 목적함수는 NLI task들과 QQP같은 큰 데이터셋에서 도움이 되지만 작은 데이터셋에서는 그렇지 못한다고 주장
- Transformer의 구조를 2048 unit의 LSTM으로 대체하였는데 MRPC를 제외하고 평균점수가 5.6점이 감소
- pre-train 없이 task를 진행하였을시 14.8% 성능이 감소

6. Conclusion

- 기존 방식과 달리 task별로 architecture를 설계해야 하는 것이 아닌 generative pre-training 모델과 discriminative fine-tuning모델을 제안 -> task-agnostic model